

A Holonic Model Of System For The Resolution Of Incidents In The Software Engineering Projects

Horacio Paggi Straneo
Facultad de Ingeniería
Universidad ORT Uruguay
Montevideo, Uruguay
horacio.paggi@gmail.com

Fernando Alonso Amo
Facultad de Informática
Universidad Politécnica de Madrid
Madrid, España
falonso@fi.upm.es

Abstract - The need of automation in the resolution of the incidents that arise in the different phases of the software Engineering Projects, the desire of to manage the knowledge about how to solve an incident, the high specialization that appears in the different sub-domains of knowledge (security, networking, etc.), not only at individuals' level but also at organizations' one and the high rate of changes in the IT staffs, lead us to propose a model of a system for the resolution of the incidents before mentioned, based on the concepts of holon and informon.

Keywords: *holon, informon, knowledge management, incidents' resolution*

I. INTRODUCTION

The management and resolution of the Incidents that happen during the course of a Software Engineering Project is one of the activities where the proficiency of the staff and the quality of the knowledge management (KM) being made (through the recording of learned lessons as well as through the use of other artifacts of KM, such as the Incidents reports, good and better practices, and organizational memories) becomes more apparent. Given that the flexibility with which we handle such events is, many times, crucial [1], in the sense of being able to react in the most efficient way to the "turbulences" associated to unpredicted changes in the priority and scheduling of resources, it turns out to be necessary a management of these events that allows to have a flexible, agile and intelligent process, no matter the quantity or complexity of the parts responsible of their solution. Because of that, this work shows a model of system for the management of these incidents, which designs and executes a series of steps for their resolution, using holons and informons, and is compared with the "classical" (and most common) approach. It should be noted that, even though applications based on holons are quite common in certain industries such

as manufacturing [2] [3] [4, 5]), they are not so common in services based industries, such the Information Technology (IT) one.

This work is organized as follows: next, a brief introduction to the basic concepts used is given, then, in the Section III we comment the related work already done in the field continuing with the description of the proposed solution in the Section IV and the main conclusions drawn together with a few lines of future work in the Section V. The acknowledgments and references complete the work.

II. HOLONS AND INFORMONS

A. Incidents, taxonomies and incidents taxonomies.

In this paper we will take as definition of incident the one proposed by Cobit 4.1 [[6] pág. 189] which is the same that the given by ITIL [7] [8] and by the norm ISO/IEC 20000-1[[9] pág. 2]): "any event which is not part of the standard operation of the service and which causes, or may cause, an interruption or a reduction of the quality of the service".

Generally, incidents are handled by using taxonomies: they are grouped in classes (for example, the incidents of the same class being solved in the same way) and these classes are set through presumed relationships between the incidents [10]. By defining and using these taxonomies we are expliciting some type of tacit knowledge about the organization for which the taxonomy is being created (and also about the domain of knowledge - the software engineering projects -). As a consequence, this definition is implying the creation of a common vocabulary for the members of the group that is defining the taxonomy and the rest of the staff of the company (or a part of it) where it will be used and ensures, also, that there is a clear understanding of how the incidents arise [11]. The creation and use of a taxonomy would then be the "externalization" step (and may be the "combination" one if the taxonomy is created by a team) of the Nonaka and Takeuchi's SECI model [12]. Obviously, being able to exteriorize this knowledge can be very useful for the organizational KM

[13]. Additionally, understanding how the incidents are produced can help understand the type of behaviors or mistakes that causes the incidents, that is, the organizational learning is fostered. Almost the same could be said about the advantages of using a created taxonomy; even though the effort of creating the common language would be less. As a side effect, given that one knows more precisely what is each incident about, the right resources for its resolution can be assigned [12].

Hansman and Hunt [16] and Ladkin [14] propose a series of properties that a taxonomy should have in order to be considered adequate, which are basically related to an exhaustive and formally valid classification.

From another point of view, it can be said that to create an incidents' taxonomy is to apply an external semantic frame in order to interpret the system of incidents [21]. This semantic frame is created by the staff working in the organization (may be adapting existing "standard" taxonomies of the industry for that field) and, to create it, the organization and the knowledge domain where taxonomy is applied have to be known.

When the taxonomy used is "imported" from another organization many problems may appear, basically associated with the fact that the classes' structure does not reflect the reality of the recipient organization in which the incidents happen. In this case, many classes may have almost no instances while others may have too much, showing the need of an additional classification. This inadequate granularity in the taxonomy leads to poor performances in its use. Given taxonomies may be embedded in the incident management software tools available, in the form of pre-set categories.

B. Holons

The definition of holon given by Koesler [15] in 1967 as an entity "which behaves partly as wholes or wholly as parts, according to the way you look at them" is still valid although more formal definitions have been proposed [16]. Its main properties (shared with the autonomous agents [3]) are: autonomy, cooperation, adaptability, ability to react and rational behavior. Holons group in temporal hierarchies (networks) of collaboration called holarchies. When a holarchy is formed by companies, each of the possible enterprises corresponding to a different configuration of the network is called a virtual enterprise [17]. In the Software Engineering field, multi agent systems are the most common implementation of the holonic designs modeling the agents the information processing part of the holons. In a multi-agent holonic system, a holon is a concept that is leaded by the commitments between the agents to maintain a specific relation related to the goals and results in an emergent structure between them [18].

Broadly speaking, holons cooperate in a benevolent way to reach a goal; it is not a blind benevolence where

every holon spends all its time in new cooperations with the others, on the contrary, it considers constantly its commitments and duties, and it rejects to collaborate only when the required actions are impossible or highly unfavorable for the process, and, in this sense, holons can be called semi-autonomous [3].

From the multi-agent systems standpoint two forms of cooperation exist: explicit (the commitment is established through a communication) and implicit (holons are designed so the behavior oriented toward a goal emerges from the behavior of the sub-agents behavior).

C. Informons

There are two main currents in the definition of informon (from the latin "informo" – that informs- and the greek "on" –being, entity- something like "informational being" [19]) nowadays in the scientific community, one that relates it to the consciousness (where the informon is an emergent and conscious entity [19]), and a second one, followed by Sulis [11] Page 198] and Alonso et al. In this work we adopt the last point of view, more specifically: "an informon is the basic element that has meaning for a holon and that allows it to make the right decisions and to execute the proper actions" [20, 21]. An informon is an information entity and can take the form of data, news or knowledge [21]. It may have a fractal structure, in the sense that it may be considered formed by many informons, depending on the degree of generalization needed by the holon that uses it: for a certain holon a record of a certain database X will be an informon, while for another holon the whole ontology that includes that database X will be its informon. The structure of the informon (how it can be divided) is then depending on the holon and is not something intrinsic to it: for instance, for a given holon, the date "yyymmdd" is only a group of six characters while for another is the concatenation of three fields: "yy"|"mm"|"dd", that is, the structure depends on the interpretational frame adopted by every holon, which in turn would be a particular case of the frame used by the holon "higher" in the hierarchy. A holon always uses at least one informon (the description of the system state of the system in which operates) and, in turn, an informon must be used by at least one holon in order to have sense; it is a "symbiotic" relationship, or, more strictly speaking, a necessary coexistence.

Informons can be news, data, knowledge or even wisdom [21]. As long as informons are processed by holons, informons of a higher semantic level (that is, meaningful for the holons of a higher level) emerge.

III. RELATED WORK

Examples of taxonomies in the IT industry are ordinary in the case of security incidents, in [22, 23] is given an interesting review of them. More specifically, in [22] is proposed a taxonomy that uses “dimensions” (four at least) which are basically are aspects of the incidence that are used to classify it. For instance, the first dimension is (are) the category (ies) of the attack vectors: virus, worms, trojans, buffer overflows, physical attacks, etc. The second dimension is the target of the attacks (one or more targets), in the sense of the operating system and the services or applications that are affected, etc. We can see in this taxonomy an attempt of to implement the characteristic recurrence of the incidents’ structure. In [24] a classification of the incidents in the Software Engineering projects is suggested as: category (possible origins of the incident or the support staff that will solve it), central processing (access problem, problem with an application or system), network, service request (e.g., password change), etc. Maniasi gives a taxonomy for the risks classification in the Software Engineering projects [25, 26].

Regarding the holons, the concept of holon appears used previously very linked to the manufacturing industry ([27], [28], [29], [30] [31]). In the IT field many applications have been developed: a model of architecture for the management of distributed systems and services [32] and a model for the management of the health services at national level [33]. It has been used also to predict in the short run the coordination and control of multi-agent systems [34], in the management of telecommunications networks [35], in the robots design [36], etc.

The available commercial systems for incident management are usually designed to record the incidents and track their changes of status, however we have seen no intelligent systems (that is, able to find a solution from a written statement in natural language). This may be because the system would contain all the IT knowledge and experience of the company selling the system and, because of strategic considerations, it would not be sold, or maybe because of the time it would take to codify all the organizational data (that is, to install and customize the system) would make them too expensive. The development of systems that help in the resolution of security incidents has been very remarkable, perhaps because in such kind of incidents the speed of the answer is crucial. Some of the existing systems, as IRIS allow to search the solution of an incident in a knowledge base using keywords that must exist in any instance of the description field of the incident. Others, as the Cisco Security Information Management Solution allows knowledge bases with good practices, but these are information about detected new vulnerabilities that is maintained updated automatically by the provider [37].

A system for the incidents’ resolution that do not employ holons will be at best a system that describes the steps to follow and, may be, one that executes many of them (the ones that can be triggered automatically by the system, such as a program execution, the online back-up of data, etc.), however, many other actions (typically those which involve activity in the “physical” world: to unplug a cable, to add a hard disk, to buy a spare part) can only be modeled through holons, which have the possibility of having a physical process component.

IV. DESCRIPTION OF THE PROPOSED SOLUTION

A. Knowledge and incidents’ representation

The solution starts with a description of the incidents in written natural language, in Spanish, and by a process of ontology learning it generates an ontology of incidents, solutions and actions (a solution is formed by a series of actions), where exists an incidents taxonomy, a set of relationships (of association, between an incident and its solutions and between a solution and its actions, of aggregation in incident classes and solution classes and of generalization, by means of which a class is a particular case of another) and a set of functions by which to a class of incident is assigned a solution. From a practical standpoint, we say that an incidents class A is a particular case of B “if by solving A one solves B, or one is closer to solve B than one would be if A were not solved”. This will be reflected in the incidents taxonomy saying that in this case A is an “son” of B. More formally, A is “son” of B if $A \sqsubseteq B$ or if $P(A|B)=p$ or if $KA \sqsubseteq B$ where $P(\bullet|\bullet)$ and $K\bullet$ are operators extending a description logic (DL) \mathcal{ALC} [[38] page 233] in a probabilistic and epistemic sense respectively. The representation of the classes of the incidents’ taxonomy as concepts of a DL is justified by the combinatory explosion of expressions in natural language associated to a given semantic and because the methods used in the natural language processing generally require big amounts of data that generally are not available. Using DL we would represent also the solutions, the good and better practices of the domain and of the organization, and using a reasoner (such as Pellet) we could infer the solution’s validity. We have two related taxonomies: the solutions taxonomy and the incidents one.

B. *Creation of the incidents' taxonomy.*

The emergent incidents' taxonomy, as opposed to the "pre-established" ones mentioned in II.A, will be a weakly emergent entity in the Bedau's classification of emergent entities [39], built basically in a bottom-up, automatic (or semi-automatic) way, from texts (glosses) in natural language (Spanish). More concretely, it will emerge from relationships like: a) Incident A is solved with the solutions s,t,u,v ... b) Solution X is formed by the actions a,b,c,d... c) Solution S is of a type K if $s = \{a_1, a_2, \dots, a_n\}$ $a_i \in A_K \equiv$ set of actions of the class K (the solutions with the same actions form a class) d) $K \sqsubseteq Y$ if and only if $A_K \subseteq A_Y$. It can be shown that this classification of the incidents using as grouping criterion "two incidents belong to the same class if they are solved in the same way", holds the properties of determinism, repeatitibility and validity. Among the previous work done on building ontologies automatically we can cite the work of Valencia [40]: he created an application that builds an ontology without using huge quantities of initial data, extracting many inter-concept relationships such as IS-A and IS-USED-BY. The association between the linguistic expressions and the concepts that represent are stored in a knowledge base of concepts, in which we could have added, manually, many relationships (in a top-down process) to accelerate the overall process). That is, the incidents' taxonomy is created in a bottom-up/top-down process. This kind of semi-automatic learning is specifically well suited for restricted knowledge domains [41], as our case is.

C. *The informons*

Informons can be news, data, knowledge or even wisdom [21]. In the developed solution, we distinguish between news, knowledge and data to be used by a holon. For the defined holons, we distinguish between the news that the holon receives (for instance, as messages), the ones it generates (messages with a receiver already known), the source of more news (basically, input data) and the data that are changed (which, after being interpreted, will spawn the news of other holons).

D. *The holons*

In the absence of a design methodology based in holons and informons that takes in consideration the structure and semantic level of the informons associated to each holon, we used the Gaia methodology [42, 43] to

determine the existent roles and, from these roles, without using any specific methodology, the holons that should exist in the system. A methodology adequate should define a development process guided by abstraction levels providing modeling artifacts and tools to manage this process. These abstraction levels should go from top to down as well as from bottom to up (depending on the level being modeled). If the development is only top-down we obtain very rigid architectures, while a mixed process allows to obtain reconfigurable and scalable architectures [3]. This development process should include the specification of the informons structure of every holon, that is, the modeling of the semantic frame used by the holarchy to interpret the informons. This is not made by Gaia or by its relative, Roadmap. We consider two types of holons: the ones that can be decomposed in another holons and the atomic ones, the last ones considered as an agent in the case of not having a physical component. Additionally many design recommendations were adopted from the PROSA model and the ANEMONA methodology [3], making an analogy between the manufacture of products in a factory and the incidents resolution. Roles were assigned generally to different holons in a static way. If we would wish to assign them dynamically, we should take care that the characteristics of the functional decomposition (head-body) were met and that the efficacy of the holarchy is not affected by that, that the rules of the organization aren't violated (for instance, assigning two roles to a holon that never should play them concurrently) and, finally, that the problems related to the bounded rationality don't appear. Because of the lack of space, no more considerations on the holarchy organization are made here.

E. *Type of operations between holons*

From the two existent types of operation/interaction (social delegation and task delegation) we choose task delegation because it's a delegation model that can be applied to holarchies where all the agents simulate authority relationships. In this version of the model, holons have highly specialized functions and usually depend hierarchically, so the control architecture is basically hierarchic, although in many cases is heterarchic (as in the assignment of actions/solutions to external executers).

On the other hand, to designate a delegate (head holon) seemed to us as the most natural option because we are solving a problem in a distributed way: roles are defined by their sets of specific permissions and their responsibilities (norms) toward the other roles and individuals (as it is made in Gaia), and based in these norms the delegation is completely determined [44].

F. Cooperation

In our case, considering how the system was designed, the most proper model of cooperation is the implicit one. Also, the choice of the holon head is predetermined in a static way from in the design phase, and not during the execution (through negotiation between the holons), which in turn is characteristic of the hierarchic and authoritative structures.

G. Example of holon and informons

Holon name: H11

Functions and meaning

It models the set of all the external executors that coordinately execute tasks for the organization

Informons (as news) that can receive

- Request of execution of an action
- Request of execution of a solution

Informons (as news) that can generate

- Solution execution acceptance/rejection (interested/not interested)
- Action execution acceptance/rejection (interested/not interested)
- Execution result of an assigned action
- Execution result of an assigned solution

Sources of other news and generated changes

- Allocation of actions to external executors
- Allocation of Solutions to external executors
- Capacities and Solutions provided by the external executors

This is equivalent to say that the news is the informon given by the ontology of (solutions, actions, external executors) and the present state of the (allocation of actions and solutions made to external executors)

- External executors interested in executing a solution
- Result of the execution of an action
- Result of the execution of a solution
- Capacities (actions that can perform) and solutions provided
- Data of the external executors (e.g. description)
- Assigned solution
- History of the activities

That is, it generates a new version of the ontology of (solutions, actions, external executors and resources) and changes the state

Informons (as knowledge) used

- Good practices of the domain
- Good practices of the organization (quality, compliance)
- Organizational rules about allocation to external executors

H. Functioning

Figure Fig. 3a shows the connections between the holons of the highest levels, where the links between nodes always mean information and control flow. Fig. 3b depicts the roles assigned to the holons.

Given an incident (provided by the role Incident – holon F) described in natural language, the first step is to search which is the most specific concept (MSC) from which the incidence is an instance. This is made by the role RequestManager – holon E, receiving as informon the incident's description provided by the role Incident and using the incidents' taxonomy as another informon. Two roles (Resolver, Searcher – holon B) search a solution for that MSC (another informon) in the incidents' taxonomy and its relation with the solutions' taxonomy, allowing the modification of existing solutions (role SolutionModifier – holon A) by adding/deleting actions from the set of actions defined in the system (roles ActionSearcher and ActionAcceptor – holon A) or creating new actions (role ActionsManager – holon A). After a solution is determined, the role SolutionValidator (holon B) determines, using a reasoner, if it is valid. Additionally, the role SolutionAcceptor (holon B) establishes if the solution can be accepted, covering the cases of undecidability that may appear and the conditions (rules) that cannot be expressed properly with DL. Next, when a valid and acceptable solution has been found, the role RequestManager coordinates its execution (by several roles/holons: SolutionExecutor (holon G), ExternalExecutor (holon H), ActionAllocator (holon G), etc.) being able to cancel it at any moment. If the solution is normally executed, a role SolutionVerifier (holon C) verifies if the incident (problem) has been solved, and, if it hasn't, the system would return to search another solution. If all the solutions for the class A were tried, one would try with the solutions of its "sons" solutions, then with the solutions of the other classes with which A shares a "parent", etc. That is, given an incident the holon looks for a solution using analogy, checking the particular cases, etc. but not in the solutions taxonomy but in the incidents one (see Fig.1 and 2).

In the Fig. 1, incidents of the class A are solved with the solution s of type S3. A and B could be considered analogous in some way. If no solution is obtained for the class A, we'll try with the solutions for B which does not mean that the link marked as X exists, in other words, we do not try the solutions "sons" of S3 in the solutions taxonomy. The search of the solution may be repeated, as it was said, until all the available solutions are tested (or a verified solution is found), or just a number of times, number which is determined by the RequestManager. If no solution is found, the incident is passed to the role IncidentsWithoutSolutionAnalyzer that would do a solution search using deeper know ledges about the domain and the organization and with, maybe, its own ontology of incidents, solutions and actions.

Jointly with this process of search/execution/verification of Solutions, the role GoodPracticesAnalyzer maintains the rules/axioms used to validate the solutions. In this model, the roles

IncidentsWithoutSolutionAnalyzer, ActionExecuter, ExternalExecuter, RequestManager, etc. are mapped clearly to holons, more specifically; the role RequestManager is the head of the holon “Incidents Resolution System”.

The integration of external organizations in the process is allowed through the existence of the role External Executer. These holons compete for the execution of actions and whole solutions through a process of bidding.

V CONCLUSIONS AND FUTURE WORK

In this work we have outlined many aspects of a model of an incidents resolution system for the incidents that appears in the Software Engineering Projects using the idea of service-oriented holonic enterprise. The use of concepts such as holon and informon allows us to model reality in a way that we deem elegant, integrating naturally entities of different ontological levels (individuals, organizations, areas of the organization, etc.) allowing to model the physical actions involved in the incidents’ resolution – which could not be done with a software agents approach- and, because of the properties of the holons, the solutions becomes scalable and expansible. It also includes the possibility of developing many tasks (such those of the solution validation) in a mechanic or quasi-mechanic way. An example of the information used by an holon is given, in the form of input news (that comes to the holon), generated news (that arrives directly to other specific holons) and changes that the holon has made though the shared data which other holons of the holarchy will interpret to obtain the other news that affects them. For these last, only the corresponding changes in the associated data are presented. This initial description will be enhanced and formalized in the future, and shows only the actual status of a research in progress.

This work can be continued in directions:

- The study of formalisms more adequate to represent the biggest possible set of rules and restrictions in one organization and that allows, at the same time, to reason with them.
- The analysis of when and how the destruction (or lost) of the knowledge happens (for example, because of the obsolescence). In the words of the archetypal dynamics of Sulis [11], it would be the definition and implementation of “de-weaving” operators. For example, when should we eliminate the action “clean diskette unit” (and all the related solutions) from the system?
- At last, but not at least, after the resolution of many of the previous issues, the implementation of a system of

this kind would be a challenge, even if it is made for a very restricted knowledge domain (such as the security incidents). This challenge would include the definition of the criteria to be used to evaluate the performance of the designed system and to compare it with the performance of other Solutions, which could allow us to determine the contributions made by the system to the organization where it is used.

ACKNOWLEDGMENT

This research has been funded by the grant S-C-BE 55/18, Préstamo BID OCUR /1296-PDT of the Ministerio de Educación y Cultura, R.O del Uruguay.

REFERENCES

- [1] Ministerio de Administraciones Públicas de España, *Magerit - versión 2. Metodología de análisis y gestión de riesgos de los sistemas de información*. 2006.
- [2] Ryu, K., Y. Son, and M. Jung, *Modeling and specifications of dynamic agents in fractal manufacturing systems*. Computers in Industry, 2003. **52** (2): p. 161 - 182
- [3] Botti, V. and A. Giret, *Anemona a multi-agent methodology for holonic manufacturing systems*. Advanced manufacturing issn 1860-5168, ed. D.T. Pham. 2008: Springer-Verlag.
- [4] Tharumarajah, A., A.J. Wells, and L. Nemes. *Comparison of emerging manufacturing concepts*. in *1998 IEEE International Conference on Systems, Man, and Cybernetics*, . 1998: IEEE.
- [5] Brussel, H.V., et al., *Reference architecture for holonic manufacturing systems: Prosa*. Computers in Industry 1998. **37**(3): p. 255-274.
- [6] Institute, G., *Cobit 4.1 - control objectives for information and related technology*. 2007.
- [7] Office of Government Commerce, *Itil versión 2*. ITIL Versión 2, 2003.
- [8] Office of Government Commerce. *Itil version 3 glossary v01*. 2007 [cited; Available from: citeulike-article-id:2976662
- [9] *Iso/iec 20000-1:2005*. 2005.
- [10] *Diccionario merriam-webster en línea*. 2008 [cited; <http://www.merriam-webster.com/dictionary/>
- [11] Sulis, W., et al., *Archetypal dynamics*, in *Formal descriptions of developing systems*. 2003, Kluwer Academic Publishers. p. 180-227.

- [12] Nonaka, I. and H. Takeuchi, *La organización creadora de conocimiento*. 1999: Oxford University Press.
- [13] Wilson, L. and C. Frappaolo. *Implicit knowledge management: The new frontier of corporate capability*. 1999
<http://www.knowledgeharvesting.org/papers.htm>
- [14] Ladkin, P., *On classification of factors in failures and accidents*. 1999.
- [15] Koestler, A., *The ghost in the machine*. 1982: Random House. 45-60.
- [16] Sugiyama, S. *Basic concept in service science with holon*. in *IEEE International Conference on Service Operations and Logistics, and Informatics, 2007. SOLI 2007*. 2007.
- [17] Mc and G.W. Merli, *Beyond bussiness process reingeneering - towards the holonic enterprise*. 1995, New York: John Wiley.
- [18] Fischer, K., M. Schillo, and J. Siekmann. *Holonic multiagent systems: A foundation for the organisation of multiagent systems*. in *HoloMAS*. 2003. Prague.
- [19] Zeleznikar, A., *Informon- an emergent conscious component*. Informatica, 2002(26).
- [20] Alonso, F., et al. *Fundamental elements of a software design and construction theory informons and holons*. in *International Symposium of Santa Caterina on Challenges in the Internet and Interdisciplinary Research*. 2004. Amalfi, Italy.
- [21] Martínez Luaces, M., et al., *Using modular neural networks to model self-conscioiusness and self-representation for artificial entities*. International Journal of Mathematics and Computers in Simulation, 2008. 2(2).
- [22] Hansman, S. and R. Hunt, *A taxonomy of network and computer attacks*. Computers & Security (COMPSEC), 2005. 24(1): p. 31-43.
- [23] Tsipenyuk, K., B. Chess, and G. McGraw, *Seven pernicious kingdoms: A taxonomy of software security errors*. IEEE Security & Privacy, 2005. 3(6): p. 81-84.
- [24] Van Bon, J., *Gestión de servicios ti basados en itil. Una introducción*. 2006: Van Haren Publishing.
- [25] Maniasi, S., P. Britos, and R. García-Martínez, *A taxonomy-based model for identifying risks*. JIISIC, 2006.
- [26] Maniasi, S., *Un modelo para la identificación de riesgos en base a taxonomías*. Reportes Técnicos en Ingeniería del Software, 2005. 7(2): p. 36-42.
- [27] Brennan, R., et al., *Holonic manufacturing systems: A technical overview*, in *The industrial information technology handbook*, CRC Press.
- [28] Klostermeyer, A. and E. Klemm, *Multiagent-based architecture for plant automation*, in *The industrial information technology handbook*, R. Zurawski, Editor. 2005, CRC Press.
- [29] Colombo, A.W., R. Schoop, and R. Neubert, *Collaborative (agent-based) factory automation*, in *The industrial information technology handbook*, R. Zurawski, Editor. 2005, CRC Press.
- [30] Bussman, S. *An agent-oriented architecture for holonic manufacturing control*. in *The 1st International Workshop on Intelligent Manufacturing Systems, IMS-Europe 1998*. 1998. Lausanne, Switzerland.
- [31] Ulieru, M. *Qualyntel: A holonic diagnosis and quality control system*. in *INFORM '98*. 2004. Seattle, Washington.
- [32] Soriano, F.J., *Modelo de arquitectura para gestión cooperativa de sistemas y servicios distribuidos basado en agentes autónomos*, in *Facultad de Informática*. 2003, Universidad Politécnica de Madrid: Madrid. España.
- [33] Urra, P. *Reflexiones sobre la autonomía e interoperabilidad de los sistemas en la era de las redes*. in *Foro Internacional sobre Tecnologías e Información*. 2004. Cuba.
- [34] Valckenaers, P., et al., *Emergent short-term forecasting through ant colony engineering in coordination and control systems*. Design of Complex Adaptive Systems 2006. 20(3): p. 261--278.
- [35] Chou, L.-D., et al., *Implementation of mobile-agent-based network management systems for national broadband experimental networks in taiwan*, in *Holonic and multi-agent systems for manufacturing*. 2003, Springer Berlin / Heidelberg. p. 280--289.
- [36] Durna, M., A.M. Erkmen, and I. Erkmen. *The self-reconfiguration of a holonic hand: The holonic regrasp*. in *IEEE International Conference on Intelligent Robots and Systems*. 2000.
- [37] Cisco_Systems_Inc., *Ciscowork SIMS incident resolution management*.
- [38] Baader, F., et al., *The description logics handbook*. 2007: Cambridge University Press.

- [39] Bedau, M.A., *Downward causation and the autonomy of weak emergence*. Principia, 2003. 6: p. 5-50.
- [40] Valencia García, R., *Un entorno para la extracción incremental de conocimiento desde texto en lenguaje natural*, in *Departamento de Ingeniería de la Información y las Comunicaciones*. 2005, Universidad de Murcia: Murcia.
- [41] Sánchez-Cuadrado, S., et al. *Extracción automática de relaciones semánticas*. in *2da. Conferencia Iberoamericana en Sistemas, Cibernética e Informática. CИСCI 2003*. 2003. Orlando, Florida, USA.
- [42] Wooldridge, M., N. Jennings, and D. Kinny, *The gaia methodology for agent-oriented analysis and design*, in *Autonomous agents and multi-agent systems*. 2000, Kluwer Academic Publishers.
- [43] Zambonelli, F., N. Jennings, and M. Wooldridge, *Developing multiagent systems: The gaia methodology*. ACM Transactions on Software Engineering and Methodology, 2003. 12(3): p. 317-370.
- [44] Schillo, M., et al. *Towards a definition of robustness for market-style open multi-agent systems*. . in *Fifth International Conference on Autonomous Agents (AA'01)*. 2001.

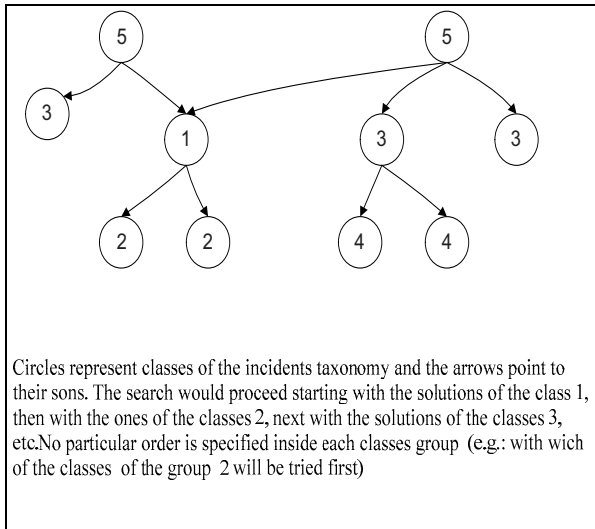


Figure 1

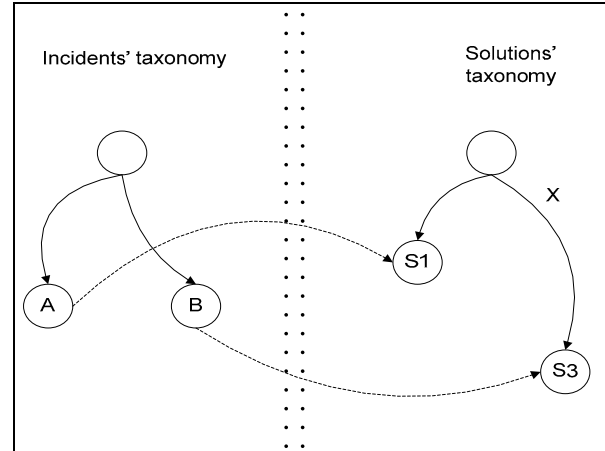


Figure 2

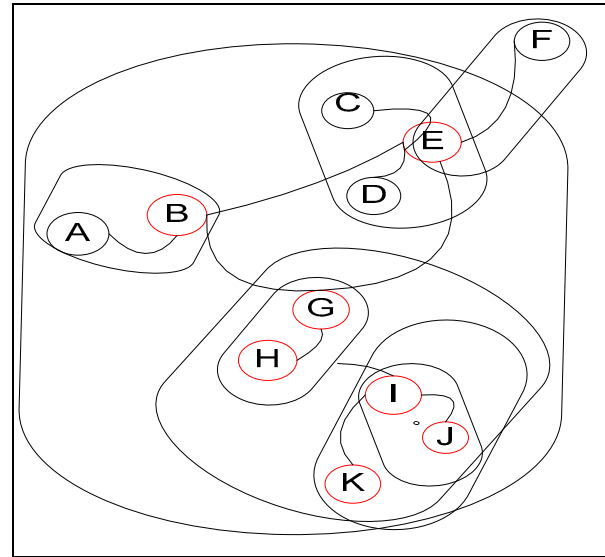


Figure 3a

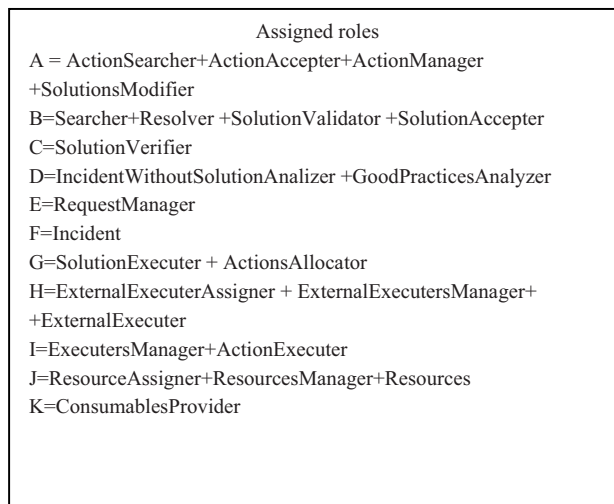


Figure 3b